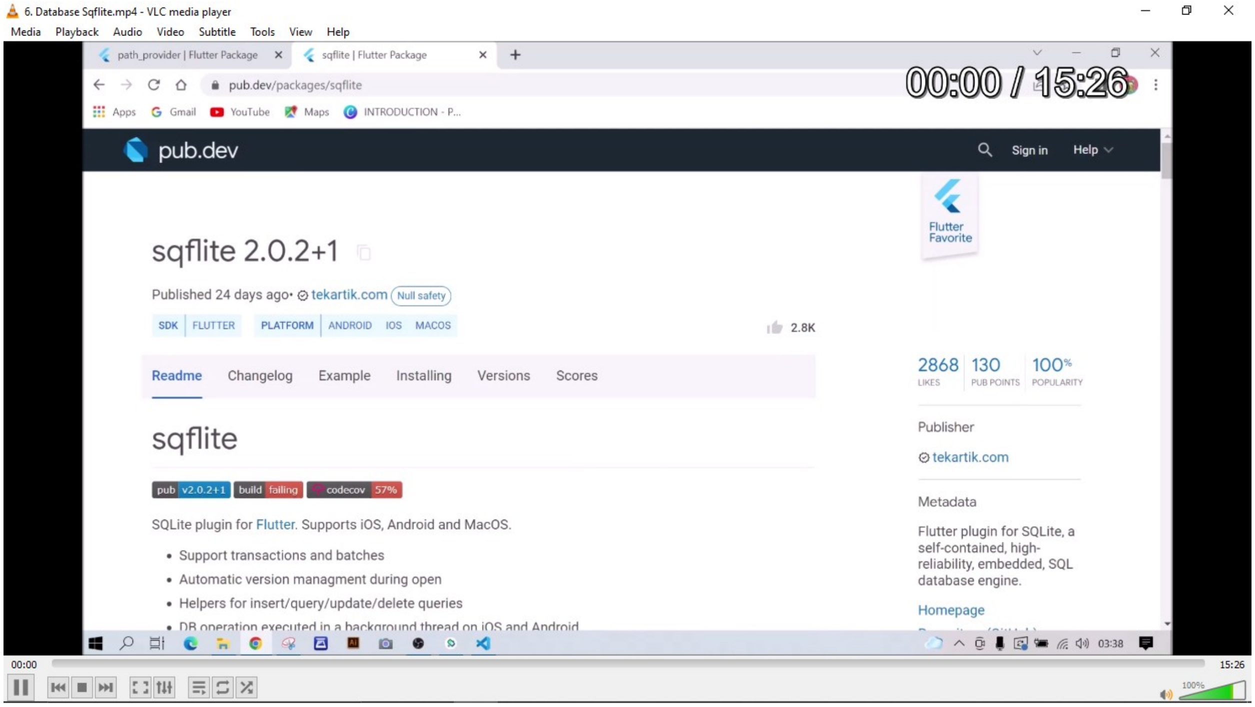
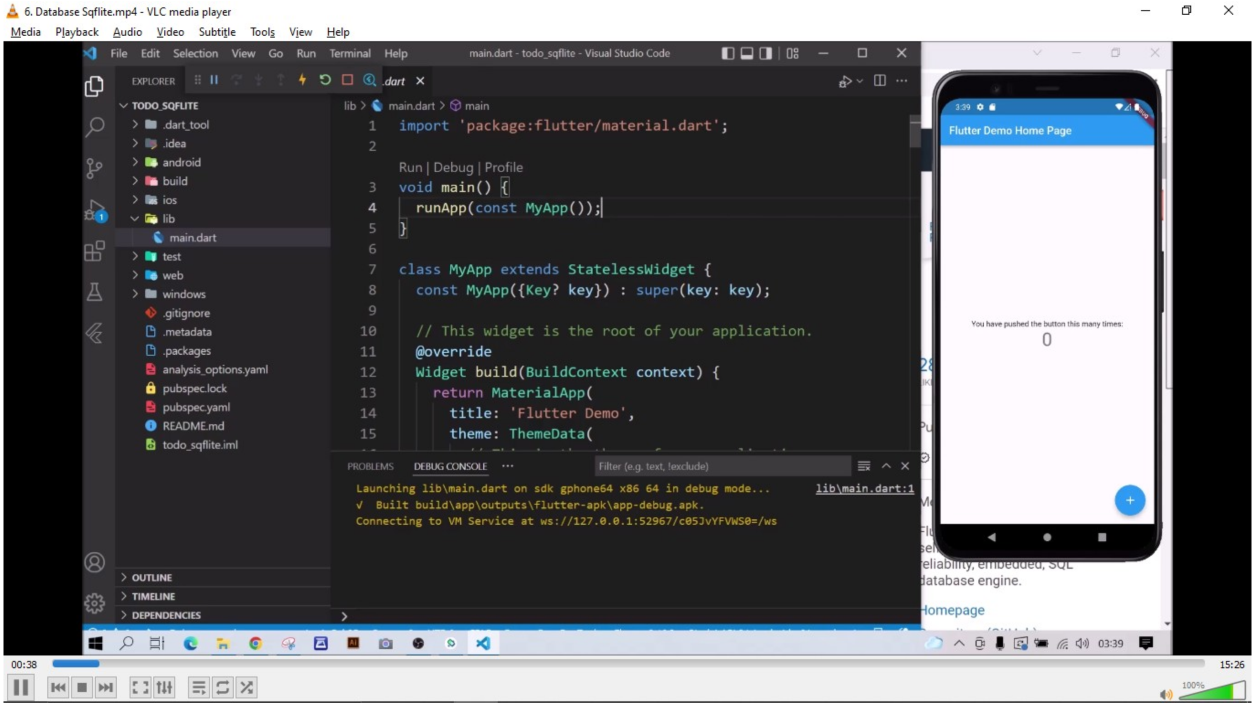
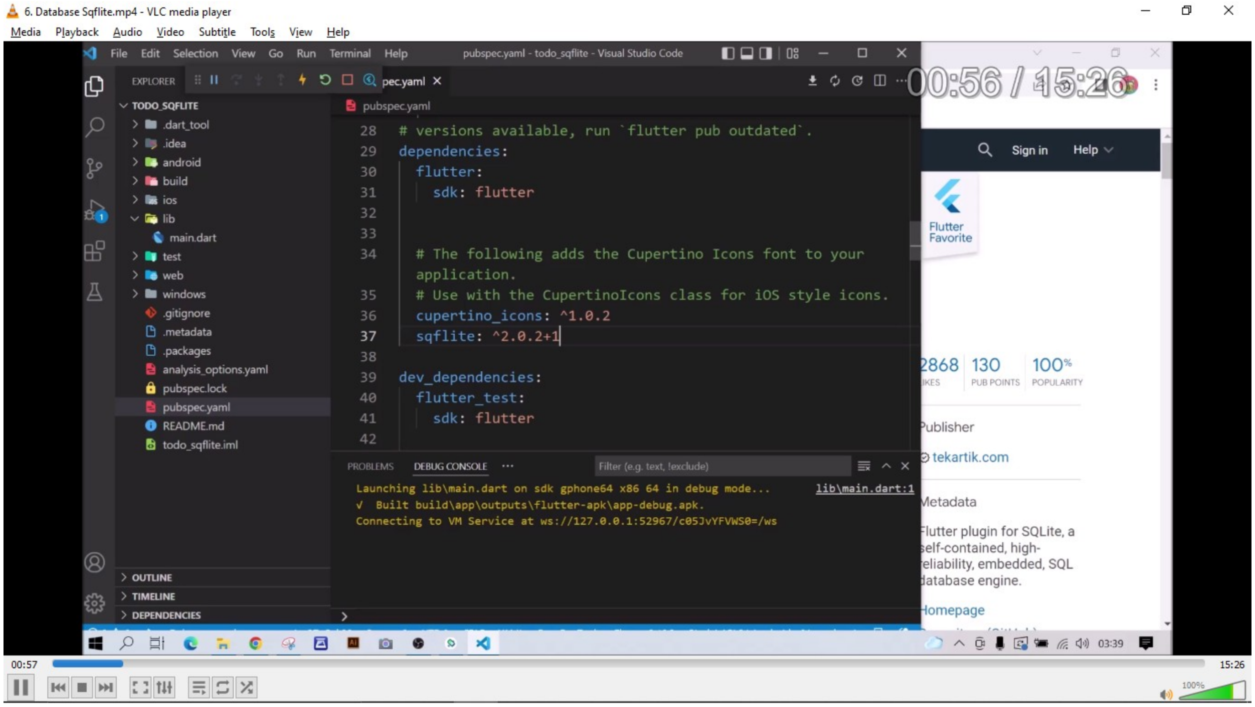
Sqflite



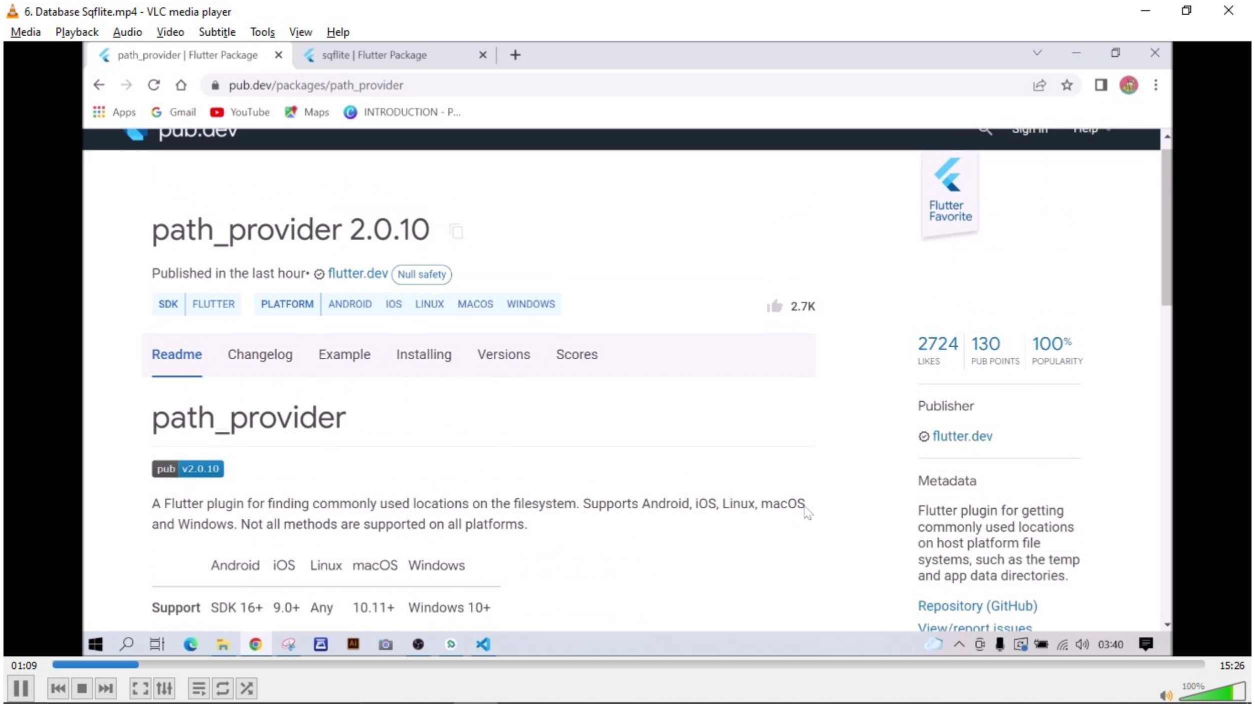
Membuat aplikasi todo app tetapi kita sudah mengunakan database
Sekalian belajar menghubungan aplikasi dgn database tapi dgn sqlite
Sebelumnya project baawan flutter



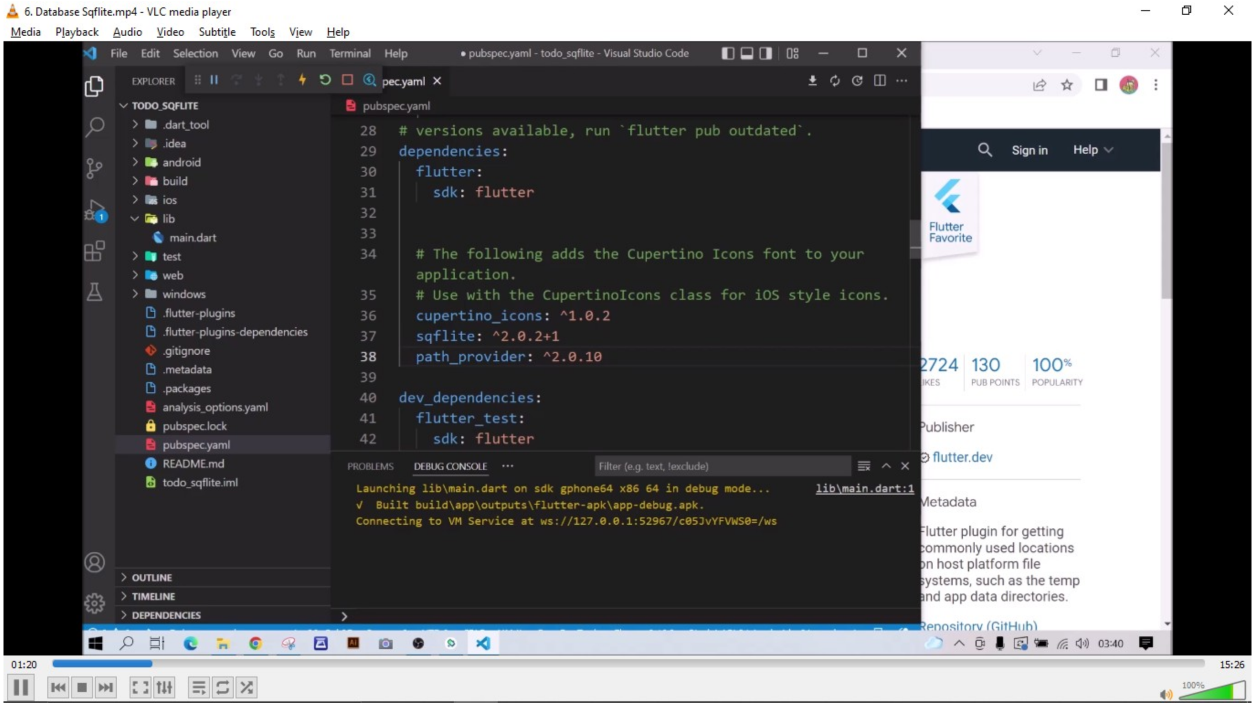
Intsal dahulu package nya



Kemudian kedua package providers



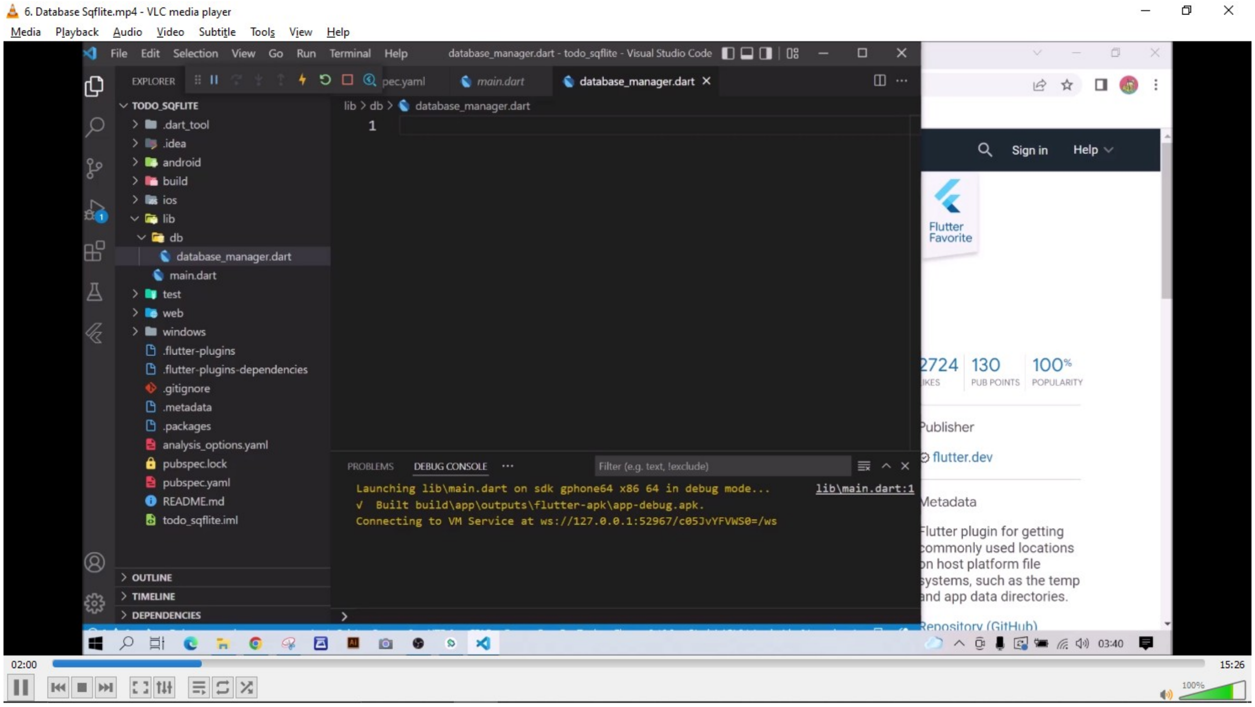
Ini sebuah package untuk mencari file system kita sebuah penyimpan ke dalam database kita



Untuk semantara main dart tidak diubah2

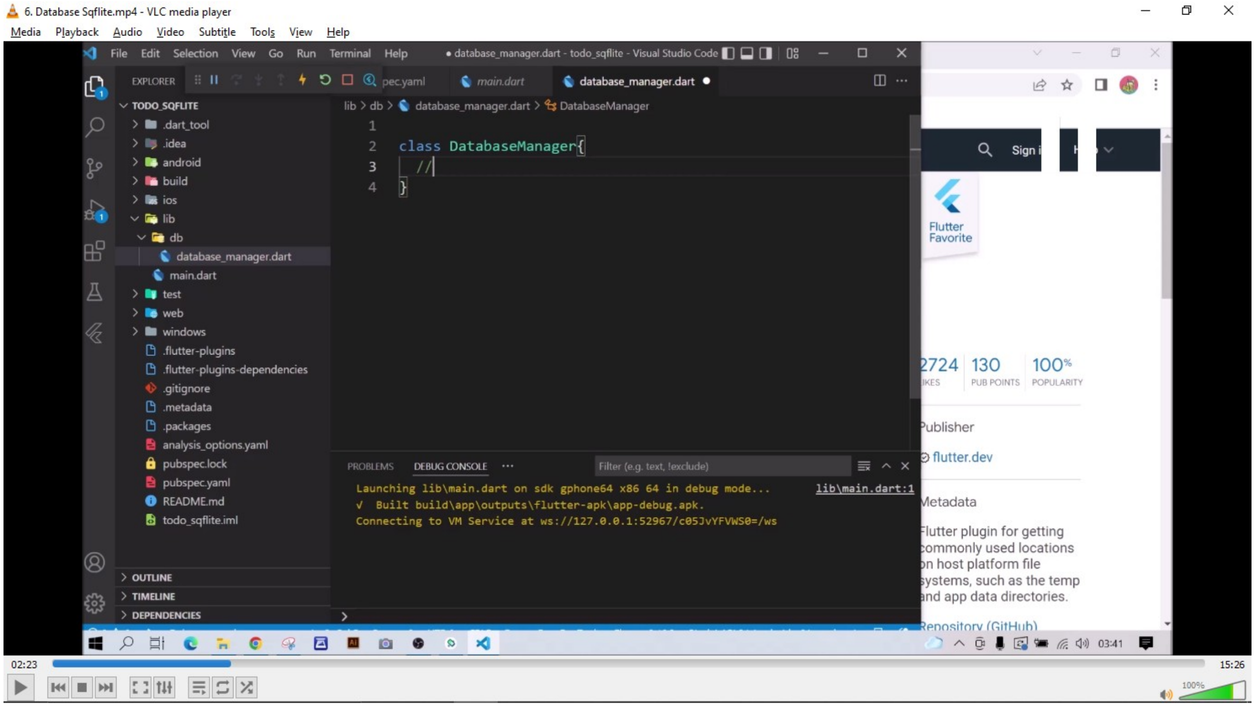
Karna kali ini hanya membuat database saja

Buat folder baru db database_manager.dart

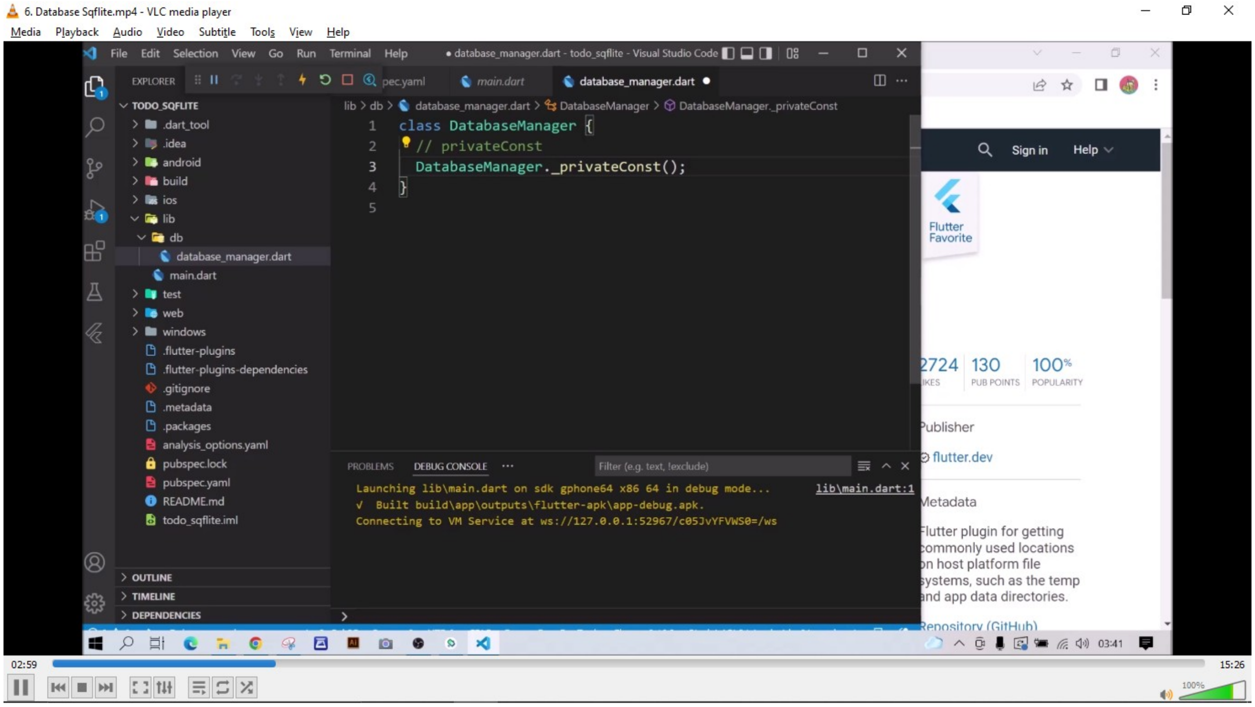


Isinya sebuah tempat untuk membuat database kita

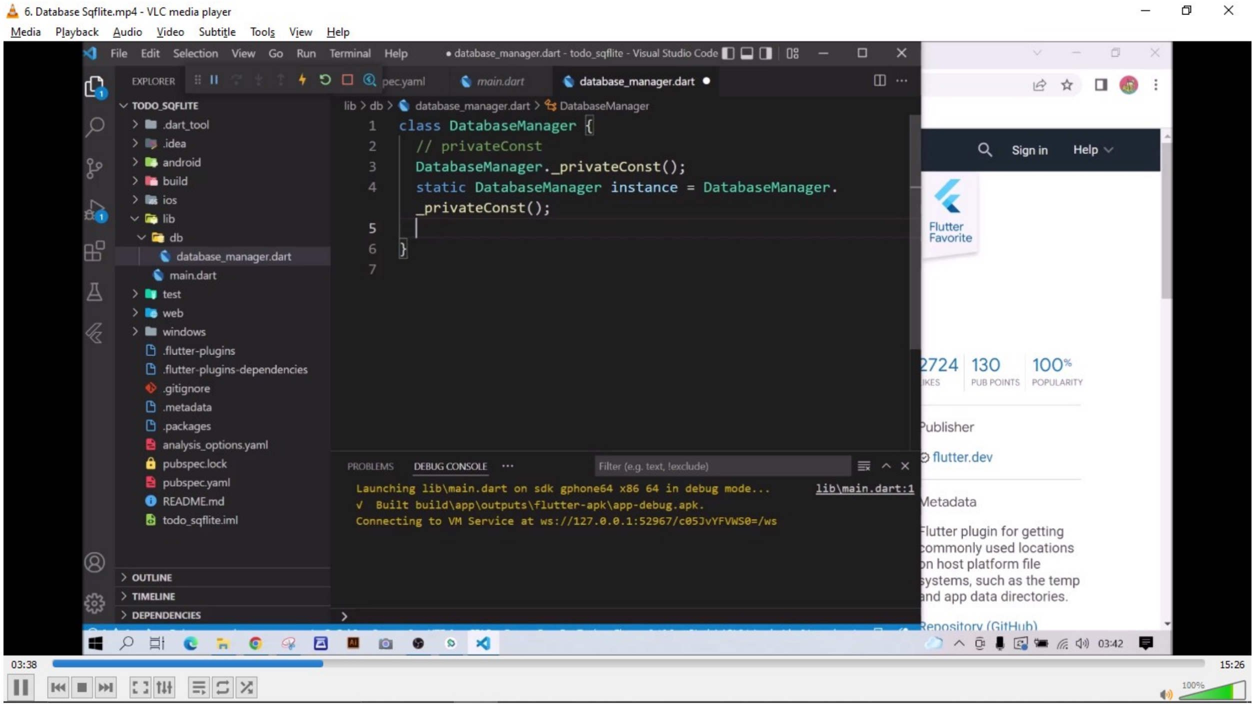
Buat class dahulu



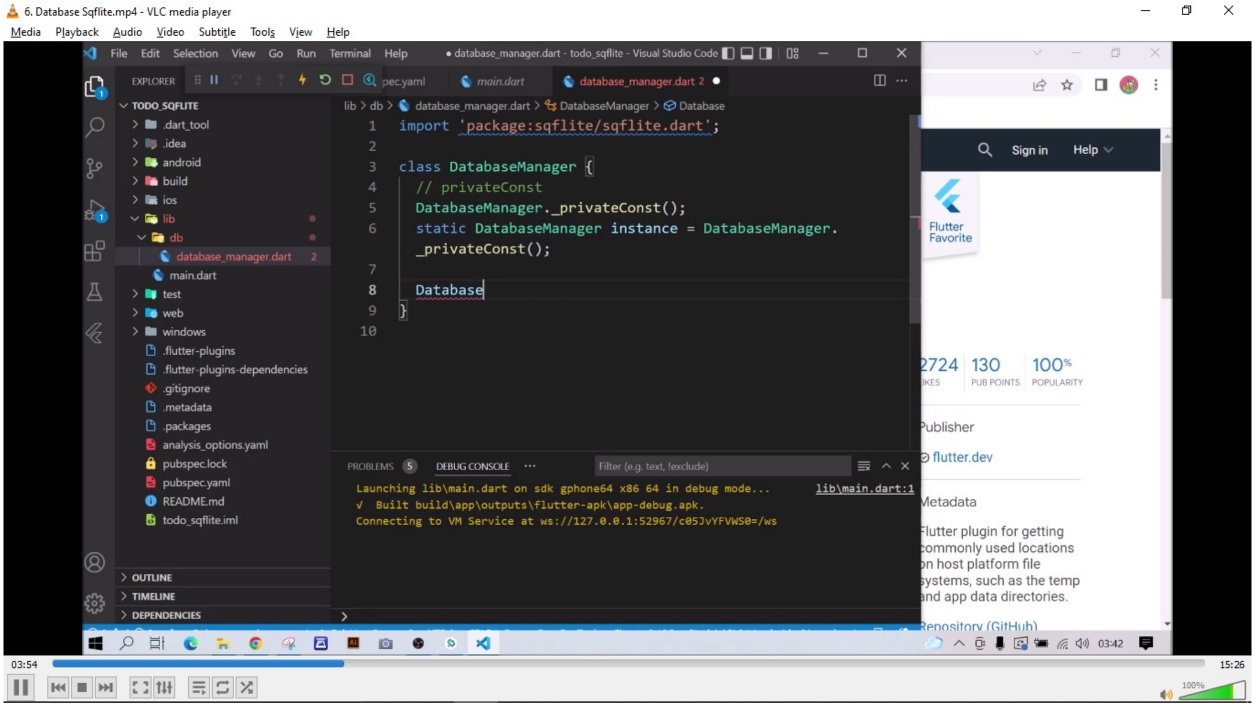
Ini class untuk databases manager
Database manager membutuhkan satu private contructor
Untuk private constructor ini tingall ketik saja



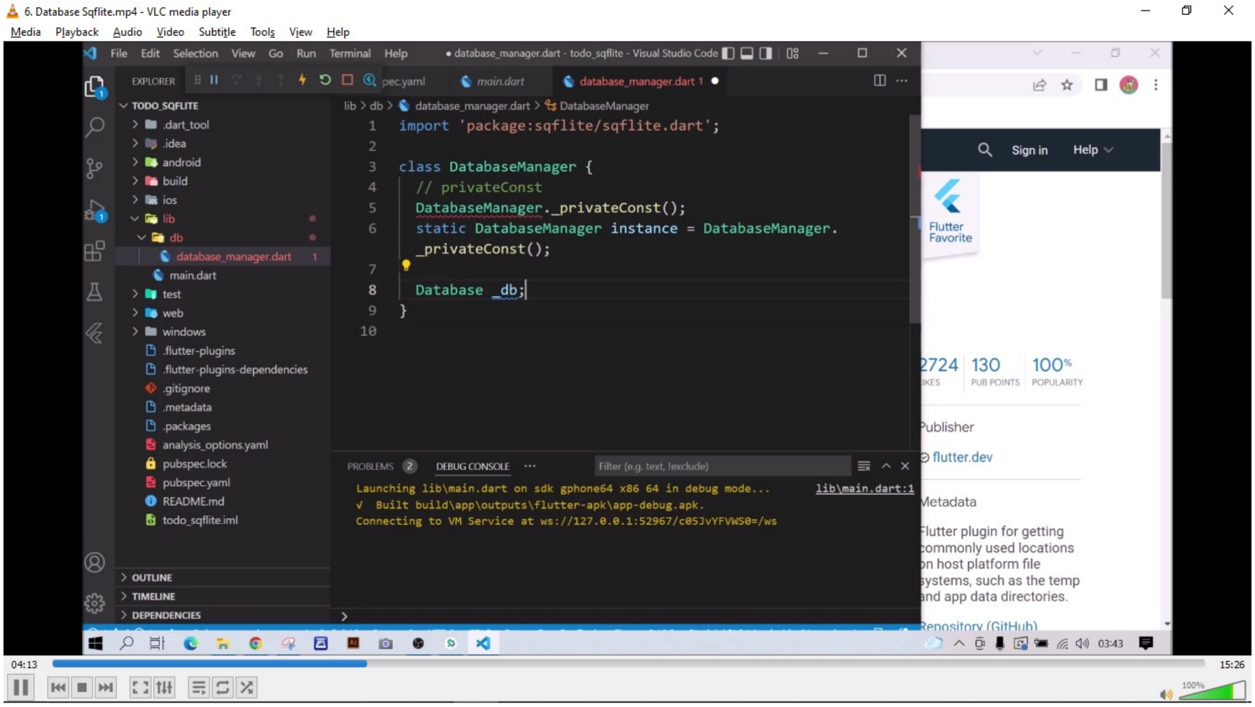
Selajutnya buat valiabel static databaseManager kasih instance untuk membuat atau memangil database kita
Isi nya memangil private constructor



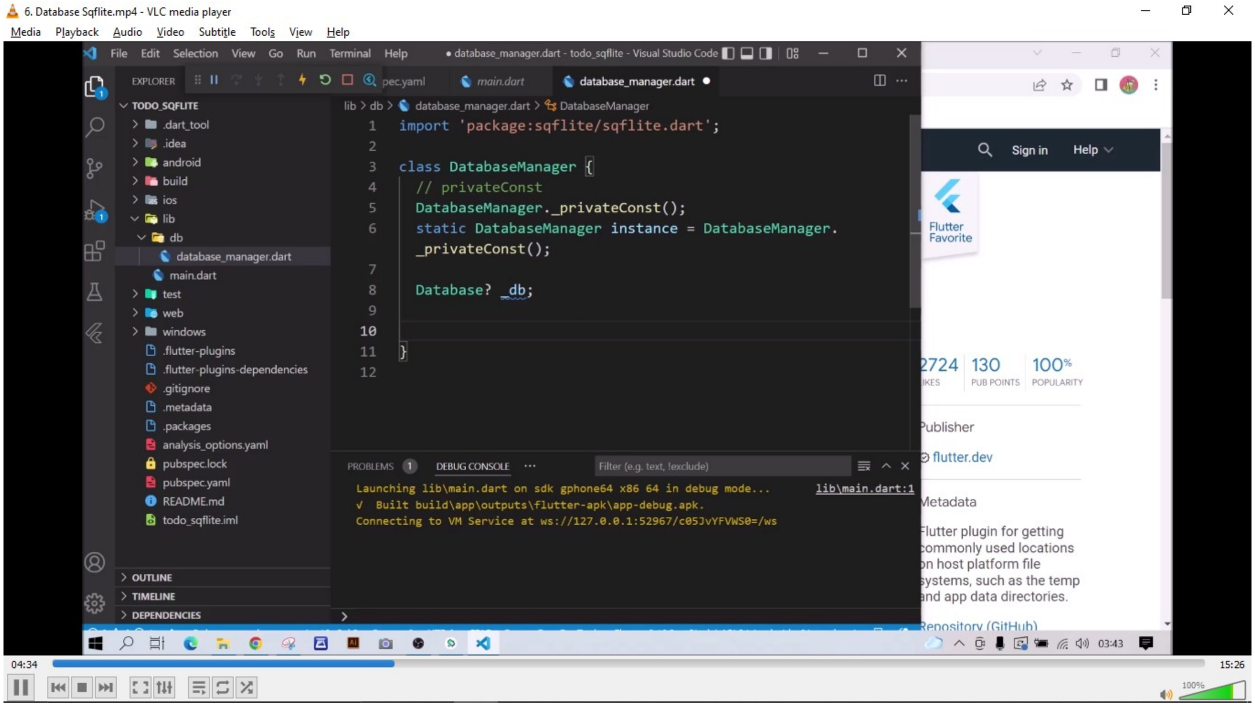
Kemudian membutuhkan satu buah lagi databasenya
Ambil dari sql lite dia automatis men import



Ini guna untuk memanggil database kita

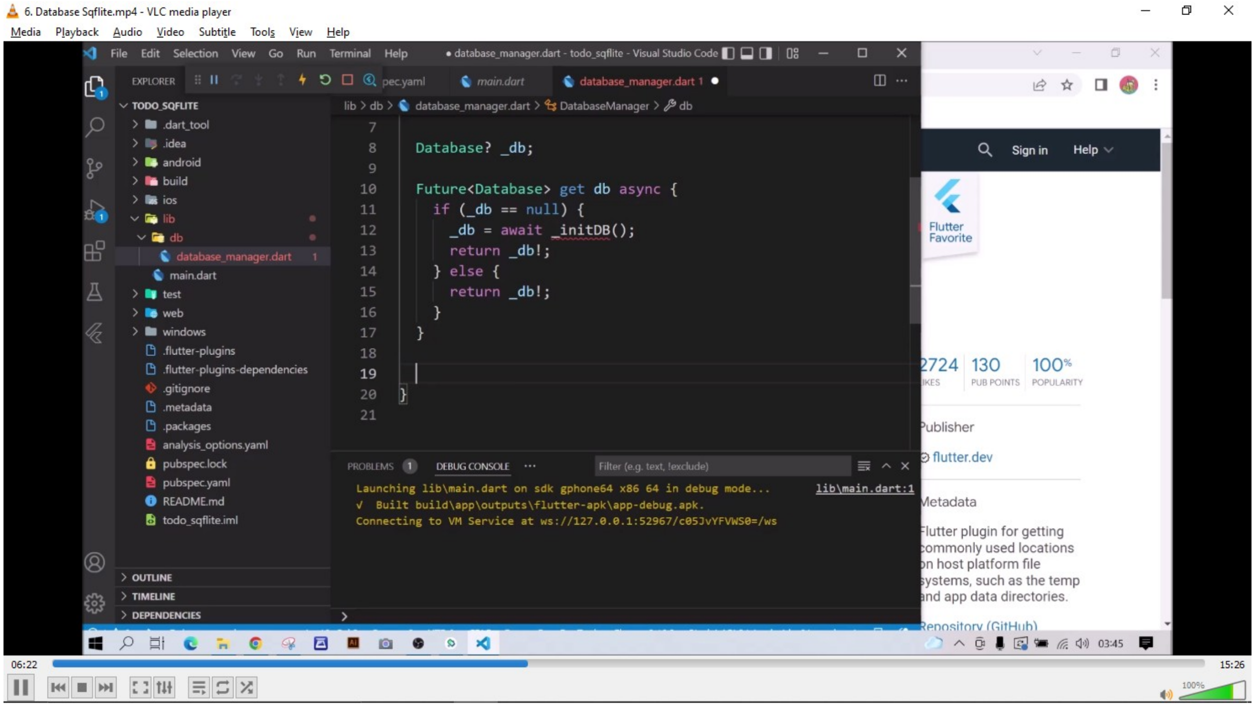


Nanti sebuah variabel nanti ini kita panggil data base kita
Karna dia null leabel kasih null? Dia boleh null

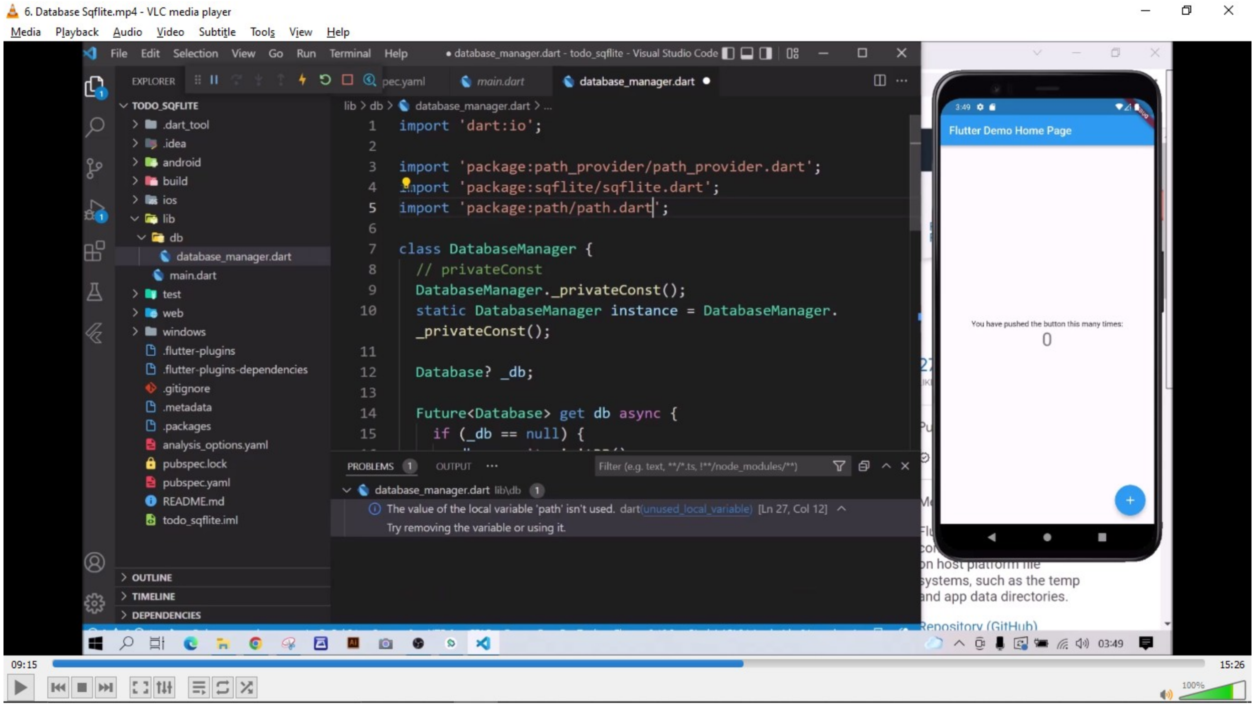


Kemudian buat get karna bentuk nya private
Buat futere dia akan mereturn database
Buat function ini untuk databse
Kasih asysn
Kita bandingkan
Jika databse tidak ada maka kita akan instisansi _db await buat initDb

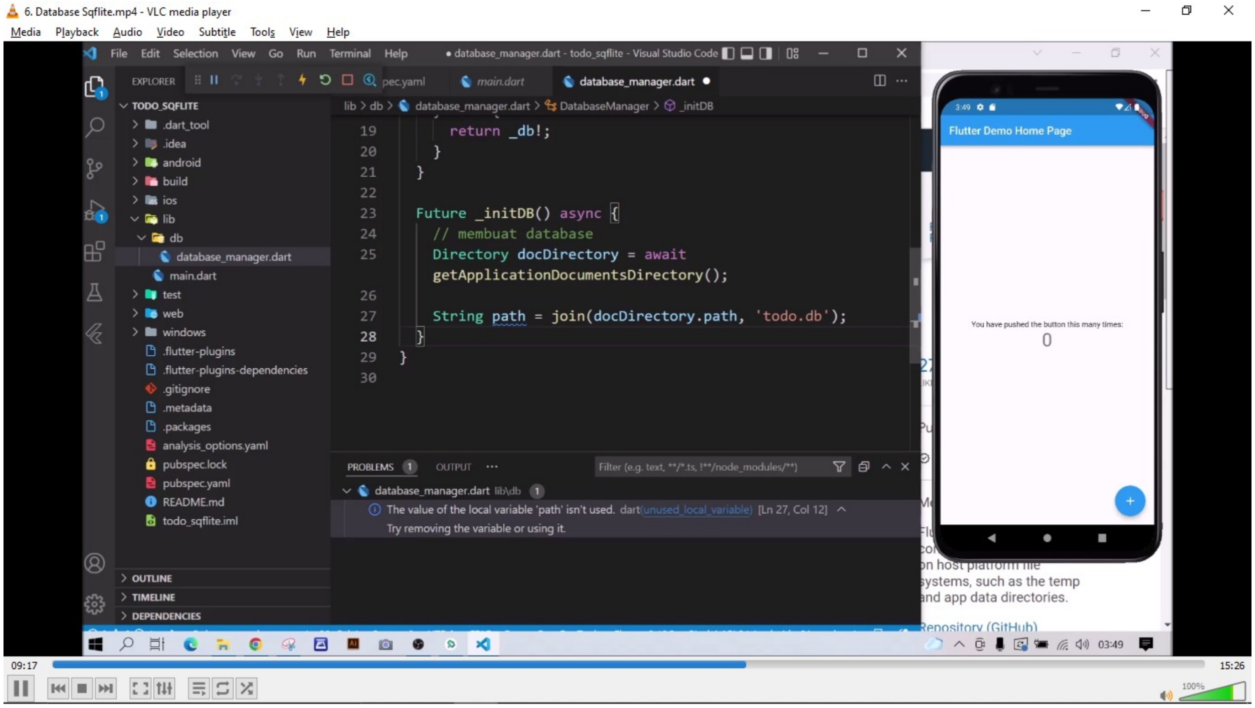
Kalau dabase belum ad akita buat ini dgn function initdb dia akan return _db
Karna dia null leabel tambah tanda seru
Kemudian kalau ada maka dia kan mereturn database nya



Kemudian buat function initDb
initDb adalah sebuah proses
tipenya future initDb jgn lupa asyn disni buat dabase nya
karna ini function initDb
pertama buat directori dahulu dia akan automatis men import dart:io
namakan dokumne directory isikan await getAplicationDocumentDirectory kita akan mengambil document dari device isinya aplikasi document dari directory
kemudia path nya bentuk nya string kita join membuthkan (docdirector.path, ‘’kemudian nama database kita berinama apa)
jadi path ini dari provider bukan dari part
maka kita import dahulu patch nya

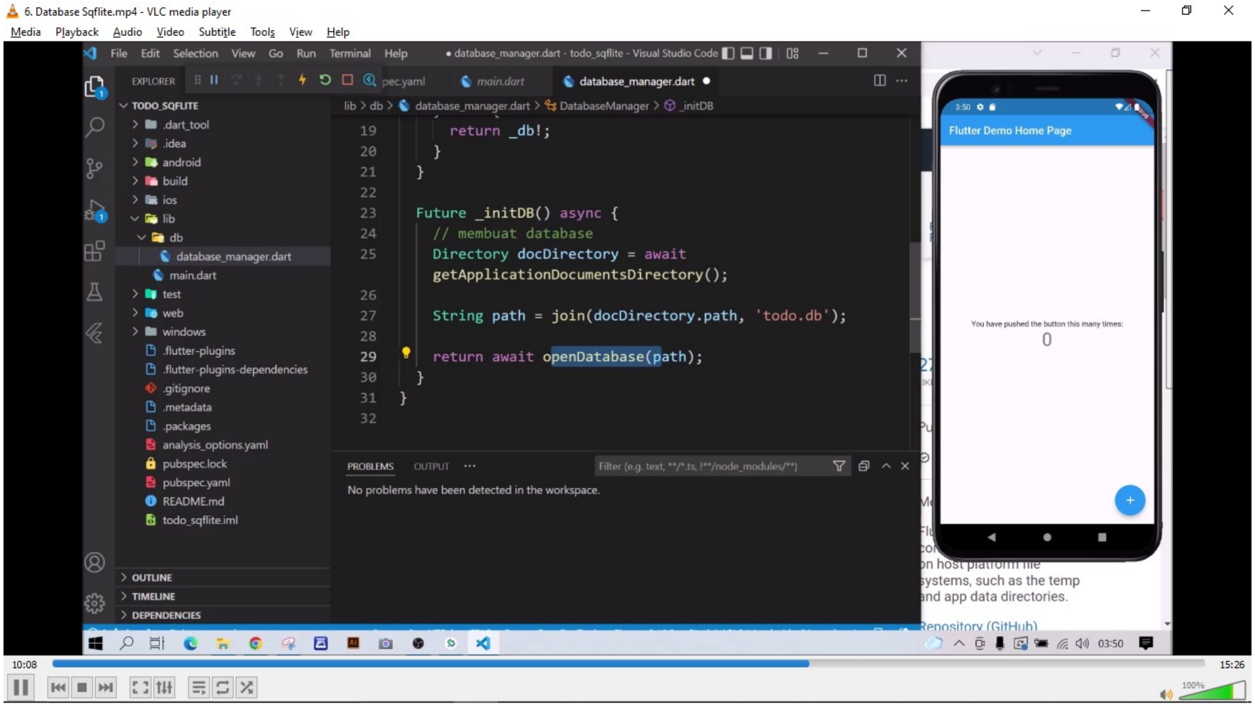


Maka tidak eror yg ini

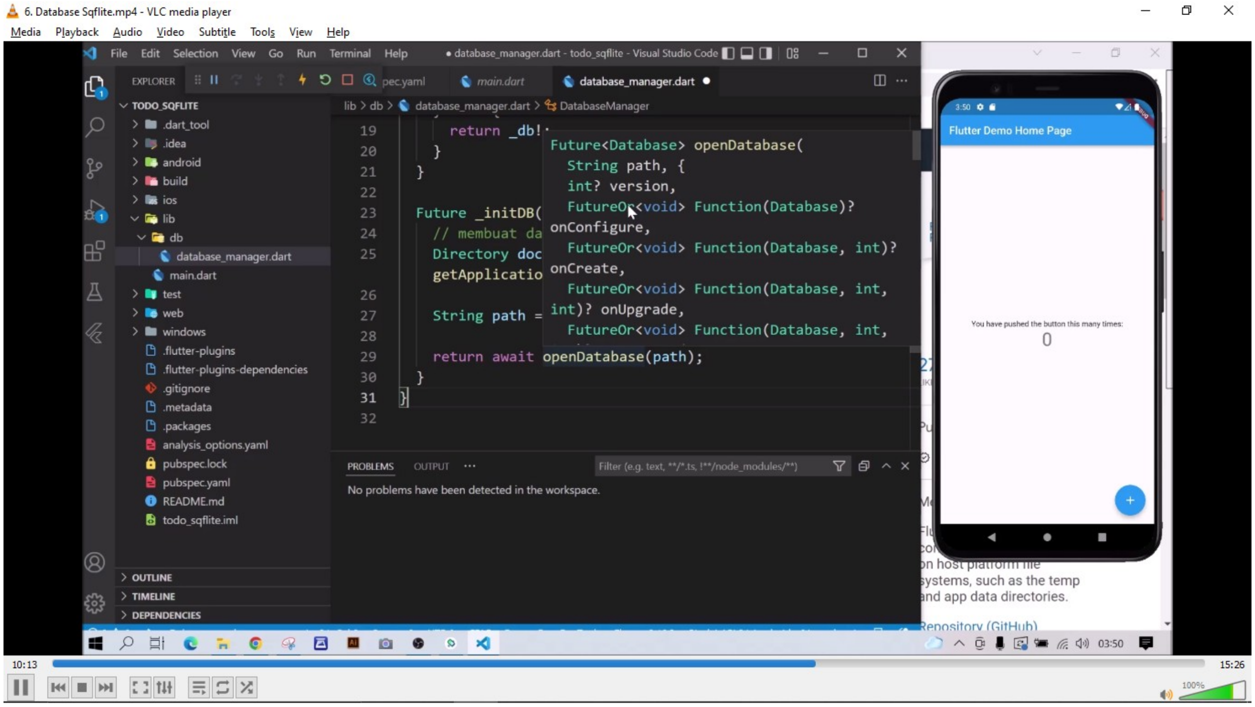


Sekarang sudah membuat databse berikutnya kita buat table nya daia akan mereturn table ini sendiri

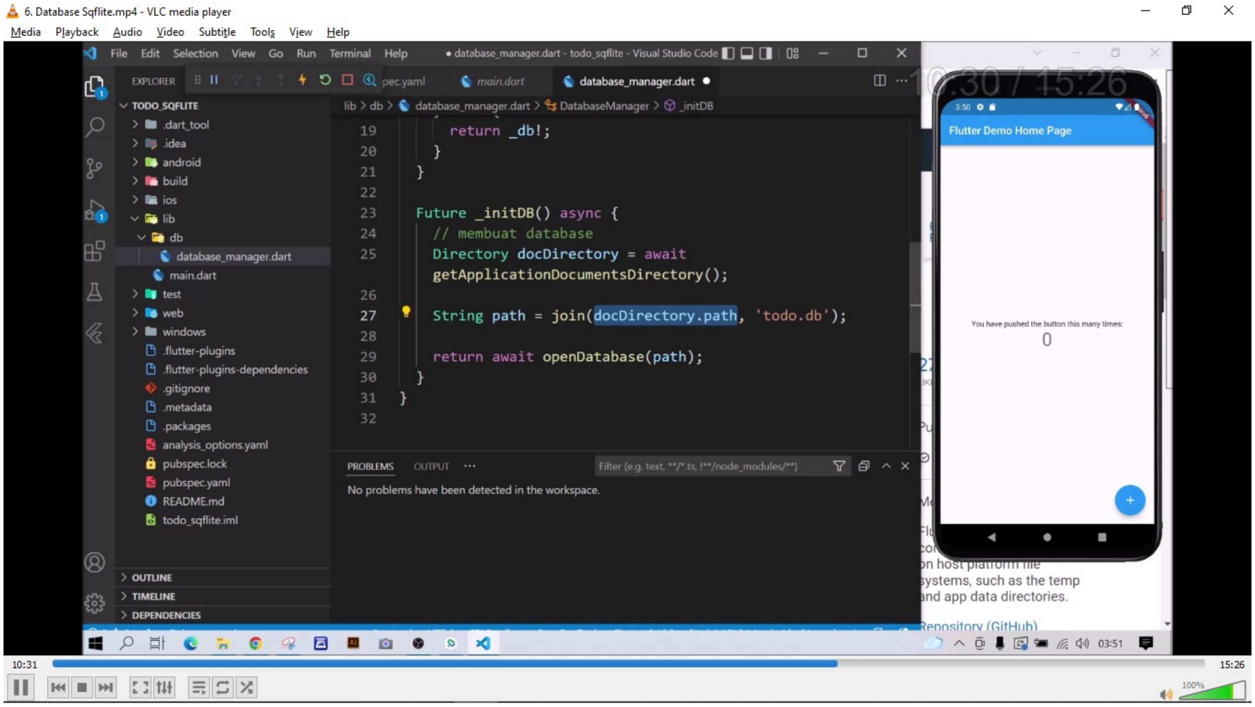
Dia buan return await openDtabase ini dari database daiakan mereturn database kita



Kemudian disni banyak parameter nya
Ada path version



Kita akan membuat tabelnya yg pertma path
Kita buat database dan folder taruh disini



Parameter kedua yaitu kasih version: 1 dab oncreate membutuhkan function kita buat dabase tulis

Database , version

Didalamnya ini buat ayscn dia aka meretur table kita

Database.execute dia akan meksekusi databse kita isinya dalah systax dari mysql

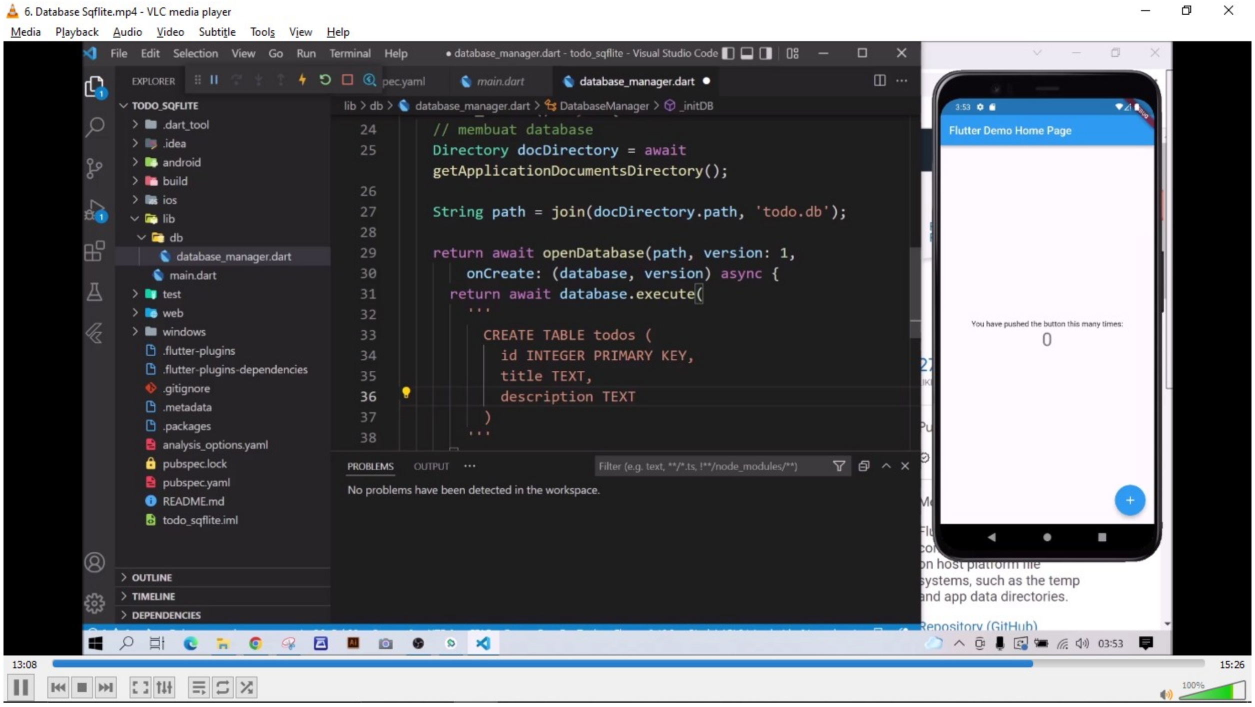
Dia mengunkan tanda petik satu tiga kali

Kita buat sytax databse kita

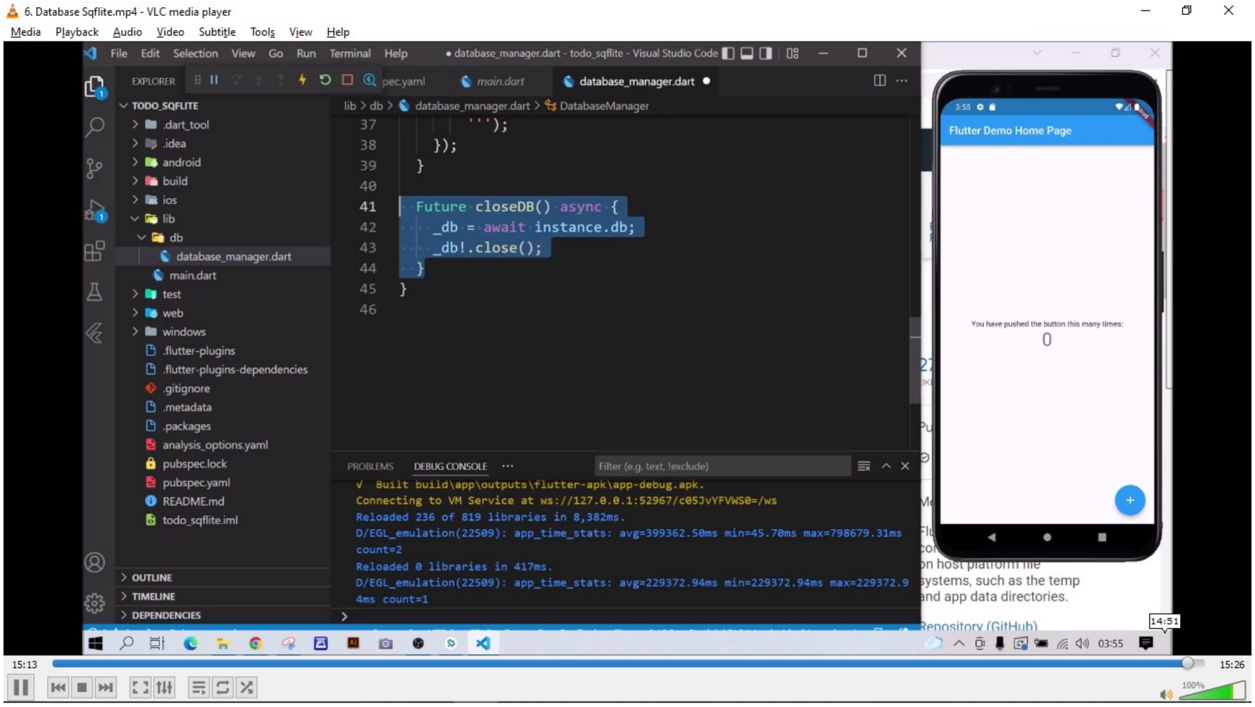
Create table apa table kita todo {

Dia butuh id Integer dan primary key yaitu tidak boleh sama

Kemudian title dand desc



}
Kita sudah selesai buat databse sudah buat table nya
Untuk pahami kita butuh
_privateConst;
Diakan get tipe nya future
Kalau belum ada databse kita buatdi function init kita join directory kita
Dan nama databse kit aitu apa
Ketika sudah buat database nya baru buat table nya
Sekarang sudah membuat database nya
Berikutnya buat future closed() guna untuk menutup database kita
Pakai async
Paka db jgn private _db ini
Ketika sudah meninstan maka dia akan harus null ! akan close
Dia akan mendrop atau menghapus database kita



Sudah mekmbuat database berikutnya menambahakan databse kita

Package

```
dependencies:
  flutter:
    sdk: flutter



  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  sqflite: ^2.0.2+1
  path_provider: ^2.0.10
```

Folder db/ database_mananger.dart

```
import 'dart:io';

import 'package:path_provider/path_provider.dart';
import 'package:sqflite/sqflite.dart';
import 'package:path/path.dart';

class DatabaseManager {
  // private const
  DatabaseManager._privateConst();
  static DatabaseManager instace = DatabaseManager._privateConst();

  Database? _db;

  Future<Database> get db async {
    if (_db == null) {
      _db = await _initDb();
      return _db!;
    } else {
      return _db!;
    }
```

```
  }

  Future _initDb() async {
    // membuat dabtabase
    Directory docDirectory = await getApplicationDocumentsDirectory();

    String path = join(docDirectory.path, 'catatan.db');

    return await openDatabase(path, version: 1,
        onCreate: (database, version) async {
      return await database.execute('''
          CREATE TABLE catat (
            id INTEGER PRIMARY KEY,
            judul TEXT,
            descripsi TEXT
          )
        ''');
    });
  }

  Future closeDB() async {
    _db = await instace.db;
    _db!.close();
  }
}
```